第 23 卷 第 4 期 华侨大学学报(自然科学版) Vol. 23 No. 4
2002 年 10 月 Journal of Huaqiao University (Natural Science) Oct. 2002

文章编号 1000-5013(2002)04-0421-06

组件式数据抽取工具的设计与实现

陈维斌 喻小光 陈启泉

(华侨大学信息科学与工程学院, 泉州 362011)

摘要 结合数据抽取处理的多源性、数据预处理功能的集成、数据抽取处理描述信息的重用等特征, 讨论数据抽取事务逻辑划分及组件包设计. 给出数据抽取包的定义、包的逻辑结构及可视化管理工具设计. 提出用元数据生成数据抽取包、用 DTS 包作为数据抽取包的执行载体的方法, 以提高数据传输性能和简化系统实现.

关键词 数据仓库, 数据抽取, 抽取包, DTS, COM 组件

中图分类号 TP 311.12 文献标识码 A

数据抽取是指从操作型数据源中获取数据, 经过清洁、变换、优化、集成等数据预处理后加载到数据仓库的过程. 数据源格式包括文本文件、XML 文档、电子表格和关系数据库等等. 数据抽取程序, 应能将分散在各个数据源中的有关某一主题的数据按主题进行合理的组织. 同时运用各种数据预处理方法[1~3], 去除源数据中大量重复和对联机分析无用的数据, 消除命名和表达方式上的不一致现象, 形成完整一致的描述, 加载到数据仓库中. 因此, 数据抽取是一项十分复杂的工作, 在数据仓库的建设和维护中, 其工作量占整个数据仓库开发工作量的 80% 左右, 对于数据仓库的建设起到了关键的作用[4]. 设计一个数据抽取工具的核心技术: (1) 数据抽取处理的可视化描述、描述信息的封装和复用; (2) 多种数据预处理功能的集成; (3) 提高大数据量的传输速度; (4) 与数据仓库系统的无缝嵌入. 本文将阐述这些技术的实现.

1 数据抽取程序的系统结构及 COM 组件包设计

1.1 数据抽取程序的主要功能

数据抽取程序提供数据抽取过程描述(包管理)、任务管理、数据预处理与加载, 以及元数据导入/导出等功能. (1) 包管理. 抽取包用来存放数据抽取过程描述信息(参见节 3.1). 包管理功能包括新建包, 即以可视化的方式描述数据抽取所需的元数据, 并籍此生成抽取包. 编辑包, 即修改包的内容. 删除包, 即取消一个抽取过程. 执行包, 即执行一个抽取过程. (2) 任务管理. 这里的任务是指将多个相关的抽取包组合在一起, 构成一个更大的可执行单元, 目的在于使零散无序的包变得紧凑有序, 以方便用户的管理. 同时, 任务可以实现一次定义, 重复使用. 任务管理包括任务的建立、修改、删除和调度等. (3) 系统元数据导入和导出. 该功能是为系统

收稿日期 2002-07-21 作者简介 陈维斌(1957-), 男, 副教授

基金项目 国家计划委员会重点项目部门专项基金资助项目

之间元数据的共享而设置的. 当一个数据源同时为两个数据仓库提供原始数据时, 我们只需针对其中一个数据仓库建立抽取包. 当另一数据仓库需要进行同样的数据抽取时, 只需将生成该包的元数据导入即可, 无需重新建立. 此外, 它还提供用户管理和查询等辅助功能.

1.2 系统结构

系统采用 3 层体系结构(图 1). 该结构使得我们能方便地将数据抽取程序所特有的数据获取、数据过滤、数据存储、数据传递等数据处理事务逻辑从客户端和服务器端独立出来, 形成相对独立的中间层(事务逻辑层). 应用 COM 技术将该层的各个主要事务逻辑设计成组件, 并且将这些组件组成一个易于组装数据抽取程序的组件包.

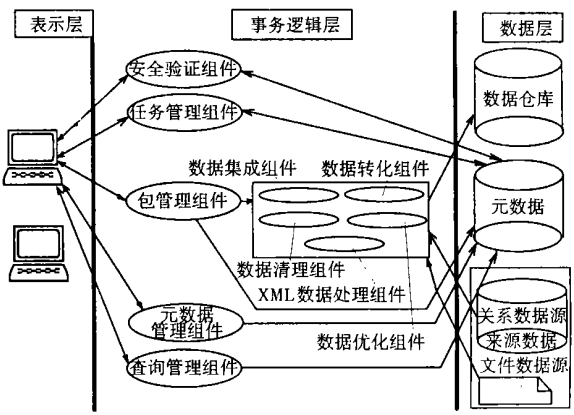

图 1 基于组件的系统结构图

1.3 主要组件简介

(1) 安全验证组件. 针对系统的不同用户, 限制或开放对元数据数据库、数据源、数据仓库的访问. (2) 任务管理组件. 将用户定义的任务信息保存到元数据数据仓库中, 设置 CreateTask, DeleteTask 等接口. (3) 元数据管理组件. 负责元数据的导入与导出, 它有 OutputMatadata 和 InputMatadata 两个接口. 其中 OutputMatadata 将元数据打包成 XML 格式文件, InputMatadata 将元数据包还原到元数据数据库中. (4) 数据综合组件. 提供 calculate(计算衍生数据)、generalize(概括) 等方法. (5) 数据清洁组件. 实现清洁功能, 提供 FillWithAConst(用常量填充), FillWithAve(用平均值填充), CheckRange(检验范围), CheckList(枚举清单) 等方法. (6) 数据变换组件. 实现各种数据变换功能, 提供 DateFormat(日期、时间格式转换), decode(字段解码), mergeField(合并字段), splitField(拆分字段) 等方法. (7) 数据优化组件. 实现各种数据优化功能, 提供 Binning-Border(按箱边缘平滑), Standardization-Avg(最大-最小规范化) 等方法. (8) XML 数据处理组件. 设置 XMLVerfy(XML 文档有效性验证), X2Rconvert(XML 模式到关系模式的转换) 等接口. (9) 包管理组件. 设置 CreatePackage, DeletePackage, ExecutePackage 等包管理功能接口, 其中 ExecutePackage 接口要调用数据综合、清洁、变换、优化等组件的方法来执行包.

1.4 元数据及数据库设计

用来支持数据抽取的元数据[4], 包括抽取工作描述、抽取工作步骤(任务)、映射规则(包括抽取表映射和抽取表的字段级映射)、数据预处理规则. 这组元数据可用来生成 1 个数据抽取包(参见节 3.2). 该包被映射成 DTS 包, 并由 DTS 完成数据抽取工作. 较之于直接利用元数据生成源代码, 本方法更易于实现并且可以获得更高的数据转换与传输性能. 此外, 利用包的封装特性和元数据导入/导出组件, 可以方便地实现元数据乃至抽取包的重用. 下面给出元数据的数据库结构. (1) DataSource 表. 保存数据源的有关信息, 包括数据源类型、数据库服务器名、数据库名、用户名、用户口令. (2) WareHouseInfo 表. 保存目的数据仓库的信息. 每个包必须对应 1 个目的数据仓库. (3) TaskTable 表. 保存抽取任务描述信息, 其中抽取任务名和包

名唯一标志 1 个记录, 执行顺序指明了包在抽取任务中的执行顺序. (4) Datamapping 表. 保存抽取表映射和抽取表的字段级映射信息. (5) DataClean 表. 保存有关数据清洁的描述. (6) DataValid 表. 保存有关数据相关性检验的描述. (7) DataIntegration 表. 保存有关数据综合的描述. (8) DataOptimize 表. 保存有关数据优化的描述.

2　数据预处理的实现方法概述

2.1　数据变换

常见的数据变换有简单变换, 日期、时间格式的转换, 由代码到名称的转换以及字段(值)拆分和字段(值)合并等. (1) 简单变换. 是指转换源数据库表中某些字段的类型、长度以及 NULL 约束. 转换工作分为 3 步: (a) 根据要求合理地定义字段的长度、名称、NULL 约束等; (b) 指定字段映射; (c) 从数据源向数据仓库加载数据. (2) 日期、时间格式的转换. 是指将来自各个数据源的不同格式的日期和时间数据变换成数据仓库的规范格式. 我们编制了一个通用的日期、时间格式转换组件. 首先, 将日期和时间字段拆成几个子部分, 再将它们合并为符合规范格式的日期和时间数据. (3) 字段值合并. 是指将源数据库中的多个字段的值合并成一个字段的值. 我们在数据变换组件中设计了函数 Merge(FieldArr() as String, TableName as String, TargetField as String). 该函数将存放在 FieldArr 数组中的字段值按顺序合并、存入 TargetField 中. (4) 字段值拆分. 是指将源数据库中的一个字段值拆分成多个字段值. 我们设计了函数 Split(Field as String, FieldArr() as String, TableName as String, Spliter as String). 它根据 Spliter 参数给出的分隔符, 将 Field 的值拆分成若干个值, 存放在 FieldArr 数组中.

2.2　数据清洁

数据清洁是一种较为复杂的数据变换, 主要用来检查字段或字段组的实际内容并去除一些非法数据, 从而保证数据的合法性和有效性. 下面简要介绍两类方法及其实现.

2.2.1　*数据合法性检查*　通过综合地应用范围检验、枚举清单及相关检验等方法制定数据清洁规则, 可以有效地进行数据合法性检查. 范围检验需要在 Select 语句中给出一个 "范围" 表达式, 枚举清单通常表示成一个集合, 用 In 操作来判别字段的值是否落在清单内. 相关检验相当于检验表之间的主码– 外键关系. 例如, Select * From Bill where Note.NoteNo in (Select Bill.NoteNo From Bill), 可以检测 Note 表的 NoteNo 字段是否与 Bill 的 NoteNo 字段相关.

2.2.2　*处理丢失的数据*　数据丢失是指某个或某些数据元没有值. 解决数据丢失问题的方法; (1) 忽略元组; (2) 用一个全局常量填充空缺值; (3) 用属性平均值填充空缺值; (4) 用与给定元组同类的所有样本的平均值填充空缺值. 我们在 "数据清洁规则描述器" 中, 提供了这些方法的选择.

2.3　数据综合

数据综合的主要任务, 是将从各种类型的数据源中得到的业务数据结合在一起, 形成新的数据. 主要手段有数据衍生、数据概括等等. 衍生数据由计算平均值、求总和或统计等多种计算手段获得, 甚至还包括复杂的业务计算的结果. 我们用 SQL 中的 sum(), avg() 等函数, 实现简单的衍生功能. 较复杂的功能则用 COM 组件实现. 数据概括是指按照 1 个或几个业务维将相近的数值加在一起, 例如商店把每日的销售额加在一起, 生成按地区计算的月销售额. 显然, 数

据概括相当于按维求和. 因此,我们用形如"Select SUM(概括字段) From 表名 Where 概括维度要符合的条件"的SQL语句来实现概括功能,对应的组件方法函数为generalize(GenField as String, TableName as String, dimension as String).

2.4 数据优化

数据优化基于某种统计或算法,将数据转换成更有利于发现本身内在规律的数据. 本系统采用去除噪声数据和规范化两种方法来进行数据优化.

3 抽取包的设计与实现

3.1 抽取包的定义及作用

抽取包是数据抽取处理的最小单位,它完整地记录了一个从数据源到数据仓库的数据导入工作的全部信息. 一个抽取包能完成两项工作. (1) 表到表的数据抽取. 可以选择来源表的全部字段,也可选择其中的部分字段. 来源字段到目的字段之间存在多对一的关系,来源字段的内容可以直接导入到目的字段中,也可以经过必要的转换后再行导入. (2) 视图到表的数据抽取. 能处理来源数据源中多个表对目的数据源中的单一表的映射. 一个包只能有一个来源和一个目的,但可以有多个任务. 这样既可以把该来源到该目的的抽取工作在一个包内全部完成,又可以减少多来源与多目的的对应关系给包的定义和执行方面带来的麻烦.

3.2 抽取包的结构

根据上面给出的抽取包的定义和功能可知,一个数据抽取包应包括3个部分. (1) 来源连接信息,包括数据源名、数据库服务器名、数据库名、用户名、用户密码等. (2) 目的连接信息,包括数据源名、数据库服务器名、数据库名、用户名、用户密码等. (3) 包任务,包括来源和目的之间的表(或视图)和字段的映射关系以及数据预处理规则描述(组织成功能脚本). 抽取包的逻辑结构,如图2所示.

3.3 用**DTS**包实现抽取包

抽取包只负责获取并保存有关数据抽取工作的描述信息,本身并不能执行数据抽取. 所以,必须将它的结构映射到一个数据传输工具中去,由该工具按抽取包所描述的任务完成数据抽取. 鉴于数据仓库的数据导入量一般都比较大,用常见的ADO, JET等工具,其效果都不理想. 而Microsoft SQL Server 2000提供的数据传输服务DTS,支持在任何具有OLE DB, ODBC等接口的数据库或规格化文本间导入、导出和转换数据. 它具有快速、准确性高、适合进行二次开发(提供了标准的COM接口)等特点,因此被选作数据传输工具.

3.3.1 *DTS包结构分析* 在DTS中,数据传输处理的描述保存在一个叫包(Package)的对象中,其结构如图3所示. 每一个包定义了包含一个或多个任务(Task)的工作流,这些任务按照一定的次序执行. 该次序被定义为一系列的步骤(Step),每个步骤必须对应包中一个任务,因此每一个包至少包含一个步骤. 任务的形式: (1) 从某个源复制数据到某个目标; (2) 使用Microsoft ActiveX脚本转换数据; (3) 在一个服务器上执行一段SQL语句. DTS包含以下几种对象. (a) 连接对象,定义每一个源和目标的OLE DB数据提供者. (b) 任务对象,定义工作项目. 其主要任务类型有数据泵任务(Data Pump Task)、执行SQL任务(Execute SQL Task)、发送邮件任务(Send Mail Task)等. (c) 步骤对象,指定任务将被执行的次序,同时也定义了一个任务的执行是否依赖前一个任务的执行结果. (d) 全局变量,用于在一个包中几个

不同的 ActiveX 脚本间传递数据或对象. (e) 转换对象, 包含了转换源和目标列的信息.

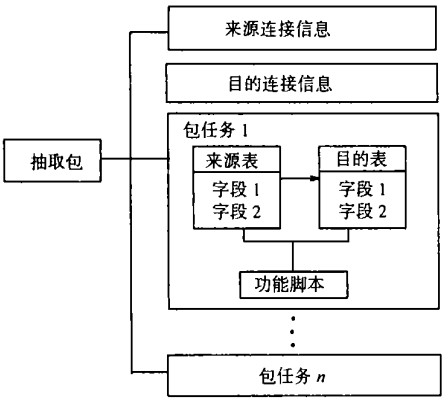

图 2　抽取包的结构图

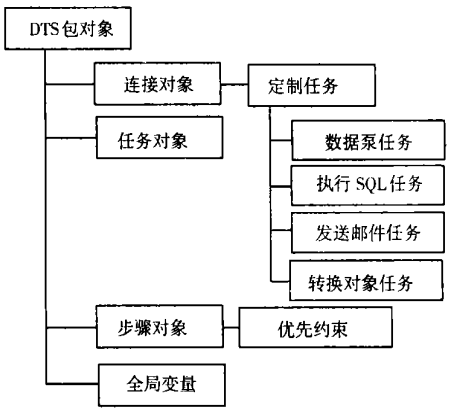

图 3　DTS 包结构示意

3.3.2　*抽取包与 DTS 对象的对应关系*　通过上面的分析, 不难看出我们设计的抽取包逻辑结构与 DTS 包的结构非常相似, 表 1 给出两者的对应关系. 利用这些关系, 可以实现由抽取包到 DTS 包的映射(参见节 3.4) .

表 1　抽取包与 DTS 对象的对应关系

抽取包	DTS 包
抽取包	Package 对象
来源连接	Connection 对象
目的连接	Connection 对象
任务	Task 对象
来源表(或视图)	来源表(或视图)
目的表	目的表
字段对应关系	字段对应关系
功能脚本	功能脚本
⋮	⋮

3.4　抽取包处理程序

该程序负责元数据的描述、保存以及抽取包的建立与执行等工作.

3.4.1　*元数据的可视化描述*　以可视化的形式, 选择数据来源和目的数据仓库. 选取对应表和字段以及指定对应字段之间的转换规则、优化规则、清洁规则等, 并利用脚本书写器编辑包任务的转换脚本, 进行包任务(任务 1～任务 n) 的添加. 我们设计了 "数据综合描述器 "、"任务定义 "、"脚本编辑器 "、"数据清洁描述器 "、"数据优化描述器 "等多个可视化描述工具来支持以上工作流程的实现. 这些描述器的内部处理除了将所获取的描述信息转换为元数据之外, 还必须建立各种转换规则与对应组件及组件方法的调用关系, 以便抽取包被执行时能动态绑定并执行对应组件的方法代码.

3.4.2　*元数据的保存*　将构成包的元数据, 保存到元数据数据库的相应表中. 例如将来源连接信息、目的连接信息、任务描述信息, 分别保存到 datasource, WarehouseInfo, ColumnTable, ScriptTable 等表中.

3.4.3　*抽取包的建立与执行*　调用包管理组件的 CreatePackage 方法从元数据数据库中读入包的元数据信息, 生成当前抽取包. 将抽取包转换为 DTS 包(将指定包的信息加载到系统定义的数据结构中), 然后调用 DTS 包的 Execute 方法执行包. 下面给出包的执行过程. (1) 读入指定的抽取包(将数据抽取元数据加载到支持包执行的数据结构中) . (2) 建立一个包对象实例. (3) 将抽取包的来源连接与目的连接指定给 DTS 包的 connection 对象. (4) 根据该包定义的任务的个数, 建立相应数量的步骤对象和任务对象, 并将步骤和任务对象成对地加入到 DTS 包对象中(Package 对象) . (5) 为每个任务指定源表和目的表, 并给出列的对应关系以及

转换脚本.(6) 调用包对象的 Execute 方法执行包,完成抽取工作.

4 结束语

我们采用 COM 技术成功地实现了节 1.3 给出的组件包,基本解决了多种数据预处理功能的集成问题.应用可视化描述技术、元数据管理技术以及 DTS 技术,实现了节 1.4 和 3.4 给出的由元数据生成抽取包、借用 DTS 对象作为抽取包的执行载体的方法.在此基础上,快速地实现了数据抽取程序,并以嵌入方式应用在《侨务数据仓库》系统中,成为该系统的一个重要的后端工具.今后应研究并实现对数据源的增量抽取,改进对 XML 文档数据的抽取方法,利用组件的易扩充性,集成更多的数据预处理功能.

参 考 文 献

1 Han J, Kamber M 著. 数据挖掘概念与技术[M]. 范 明等译. 北京: 机械工业出版社, 2001. 70～78
2 朱 焱. 浅论数据抽取、净化和转换工具[J]. 计算机应用, 2000, 20(4): 1～3
3 琚春华, 凌 云, 王光明. 基于知识的多数据源 DSS 的数据抽取技术研究[J]. 小型微型计算机系统, 2001, 22(9): 1 096～1 098
4 张维明. 数据仓库原理与应用[M]. 北京: 电子工业出版社, 2002. 90～93

Design and Implementation of a Component Type Tool for Extracting Data

Chen Weibin Yu Xiaoguang Chen Qiquan
(College of Info. Sci. & Eng., Huaqiao Univ., 362011, Quanzhou)

Abstract Combining with such characteristics of data as multi-source of their extraction and processing, integration of their pre-processing function, and reusing extraction, processing and description of information, a discussion is made on the logical devision of data extract affairs and the design of component package. On this basis, the authors give the definition of data extract package, logical structure of package and the design of visualized management tool; and put forward the methods of use metadata to generate data extract package, and use DTS package as executive carrier of data extract package, by which the performance of data transmission can be improved and system implementation can be simplified.

Keywords data warehouse, data extraction, extract pakage, DTS, COM component